1012 - Guilty Prince

Link submit: http://lightoj.com/volume_showproblem.php?problem=1012

Link solution:

C++: http://ideone.com/xiNVED

python: https://ideone.com/iSewBz

Tóm tắt đề:

Cho một bảng hình chữ nhật kích thước W x H, các ký tự trên bảng gồm 3 loại ký tự thể hiện như sau:

- Loại '.' : Đất
- Loại '#': Nước
- Loại '@': Vị trí hiện tại đang đứng.

Yêu cầu: Hãy đếm có bao nhiêu ô '.' (bao gồm cả ô có chứa ký tự '@') có đường đi tới ô '@'. 2 ô có thể đi được với nhau nếu chúng có chung cạnh.

Input

Dòng đầu tiên chứa một số nguyên T thể hiện số bộ test của đề bài

T bộ dữ liệu đầu vào sau, mỗi bộ dữ liệu được tổ chức như sau:

- Dòng đầu chứa 2 số nguyên dương W và H thể hiện rằng là chiều rộng và chiều dài của bảng.
- W dòng sau, mỗi dòng chứa H ký tự chỉ bao gồm 3 loại '.' , '#' và '@'.

Output

Gồm T dòng, dóng thứ i được tổ chức dưới dạng như sau:

"Case i: ans", trong đó ans thể hiện rằng là kết quả của bộ dữ liệu tương ứng.

Example

Sample Input	Output for Sample Input
4 6 9 ....#. .....#	Case 1: 45 Case 2: 59 Case 3: 6 Case 4: 13

.... #@...# .#..#. 11 9 .#......... .#.#######. .#.#.....#. .#.#.###.#. .#.#..@#.#. .#.#####.#. .#.......#. .#########. .... 11 6 ..#..#..#.. ..#..#..#.. ..#..#..### ..#..#..#@. ..#..#..#.. ..#..#..#.. 7 7 ..#.#.. ..#.#.. ###.### ...@... ###.### ..#.#.. ..#.#..	

Hướng dẫn giải:

Nhận xét: Bạn có thể nhận xét rằng 2 ô (x , y) và ô (i , j) sẽ đi được với nhau nếu chúng có chung cạnh. Do đó, nếu bạn có thể xem xét rằng mỗi ô trên bảng là một đỉnh của một đồ thị, thì 2 đỉnh trên đồ thị sẽ có một cạnh nối với nhau nếu chúng thỏa mãn 3 tính chất sau đây:

- 2 ô được biểu diễn bởi 2 đỉnh của đồ thị phải nằm trong bảng đầu vào.
- 2 ô không ô nào được phép biểu diễn bởi ký tự '#'.
- 2 ô phải chung cạnh với nhau.

Như vậy, sau khi ta xây dựng được một đồ thị vô hướng như vậy rồi, ý tưởng của ta sẽ thật đơn giản như sau:

- Ta xác định vị trí của ô '@', giả sử là ô (sx , sy).
- Với mỗi ô (i , j) trong bảng mà được biểu diễn bằng dấu '.', ta sẽ sử dụng kỹ thuật duyệt DFS từ ô (i , j) để có thể đến được ô (sx , sy) hay không. Nếu có, ta tăng biến kết quả ans lên 1.
- Kết quả là ans + 1 (Tính cả ô (sx , sy))

Đánh giá độ phức tạp thuật toán :

- Độ phức tạp thời gian: O(W * W * H * H)
- Độ phức tạp không gian: O(W * H)

Vì W và H không vượt quá 20 nên độ phức tạp O(W * W * H * H) là một độ phức tạp hoàn toàn có thể chấp nhận được.

Cải tiến:

- Ta có thể cải tiến thuật toán ở trên dựa vào một nhận xét rất tự nhiên như sau: Ta thấy rằng mọi ô giả sử có đường đi đến ô (sx , sy) thì chúng đều sẽ tập trung lại tại ô (sx , sy). Do đó, nếu như ta có thể nhìn nhận bài toán ở một khía cạnh khác, chúng ta sẽ phát hiện rằng: Thay vì với mỗi ô (i , j) và ta kiểm tra xem có đường đi tới (sx , sy), thì bây giờ ta sẽ xuất phát từ ô (sx , sy), ta đi loang ra các đỉnh khác, mỗi lần ta tiếp cận được với một đỉnh mà chưa được xét thì ta sẽ tăng biến ans lên 1 đơn vị. Kết quả là ans.

Đánh giá độ phức tạp thuật toán:

- Độ phức tạp thời gian: O(W * H)
- Độ phức tạp không gian: O(W * H).